Tiled with LibGDX

WHAT IS TILED?

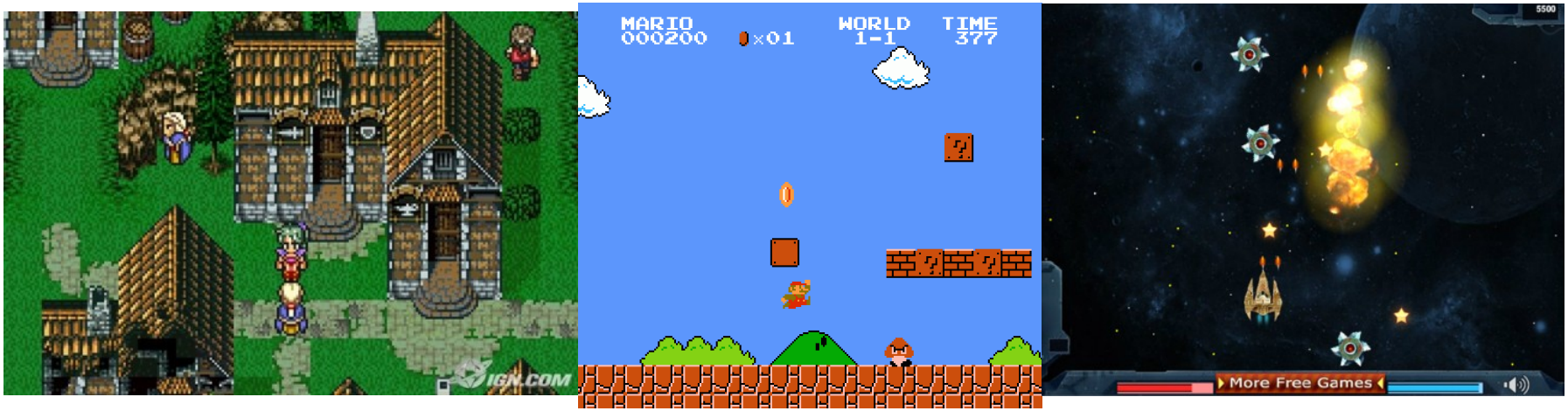
And Why We Should Use It?

- **Tiled is a free software 2D level editor**.
- Supports orthogonal, isometric and hexagonal projections.
- The author accepts voluntary payments
- Website: http://www.mapeditor.org/

WHAT IS TILED?

Examples Of Projections - Ortogonal

- It's the simplest type as there's no superposition between the tiles.
- Tiles are usually squares.
- Top-down or scrolling (horizontal and vertical).



WHAT IS TILED?

Examples Of Projections - Isometric

- Fake 3D effect
- 45 degrees perspective
- Diamond-shaped tiles
- Supeposition of the closer tiles over the further ones

WHAT IS TILED?

Examples Of Projections - Hexagonal

- Tiles are hexagons, gives six possible movement directions
- Very common in tactic war games

USING TILED

Creating A Map

- We are going to focus on the ortogonal projection, the one that the lessons 6 and 7 use.
- First of all, a tile set is needed in order to build up our map.
- Tile sets allow saving resources, something very important back in the old days.

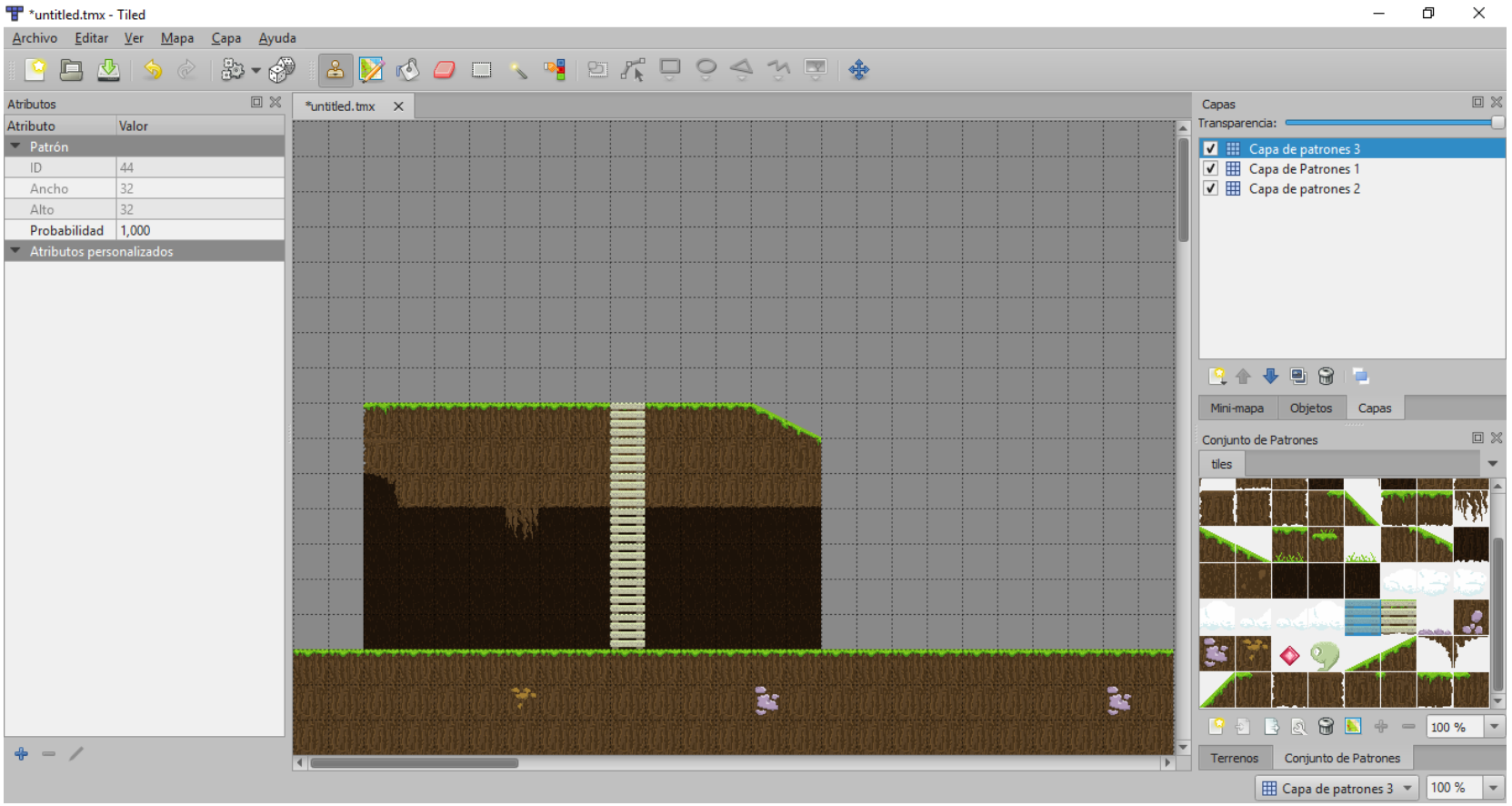
USING TILED

Creating A Map

- The technical part is quite easy, Tiled has some tools that can help you to use the tile set faster and finish the repetitive tasks faster like the stamp.
- However you need to take into account several things like the size of the characters, how long it can jump and so on.
- Some creativity is needed, that's why tile sets has decorations, to add some variety to the maps.
- Tiled allows to use several layers to add some depth.

USING TILED

Creating A Map



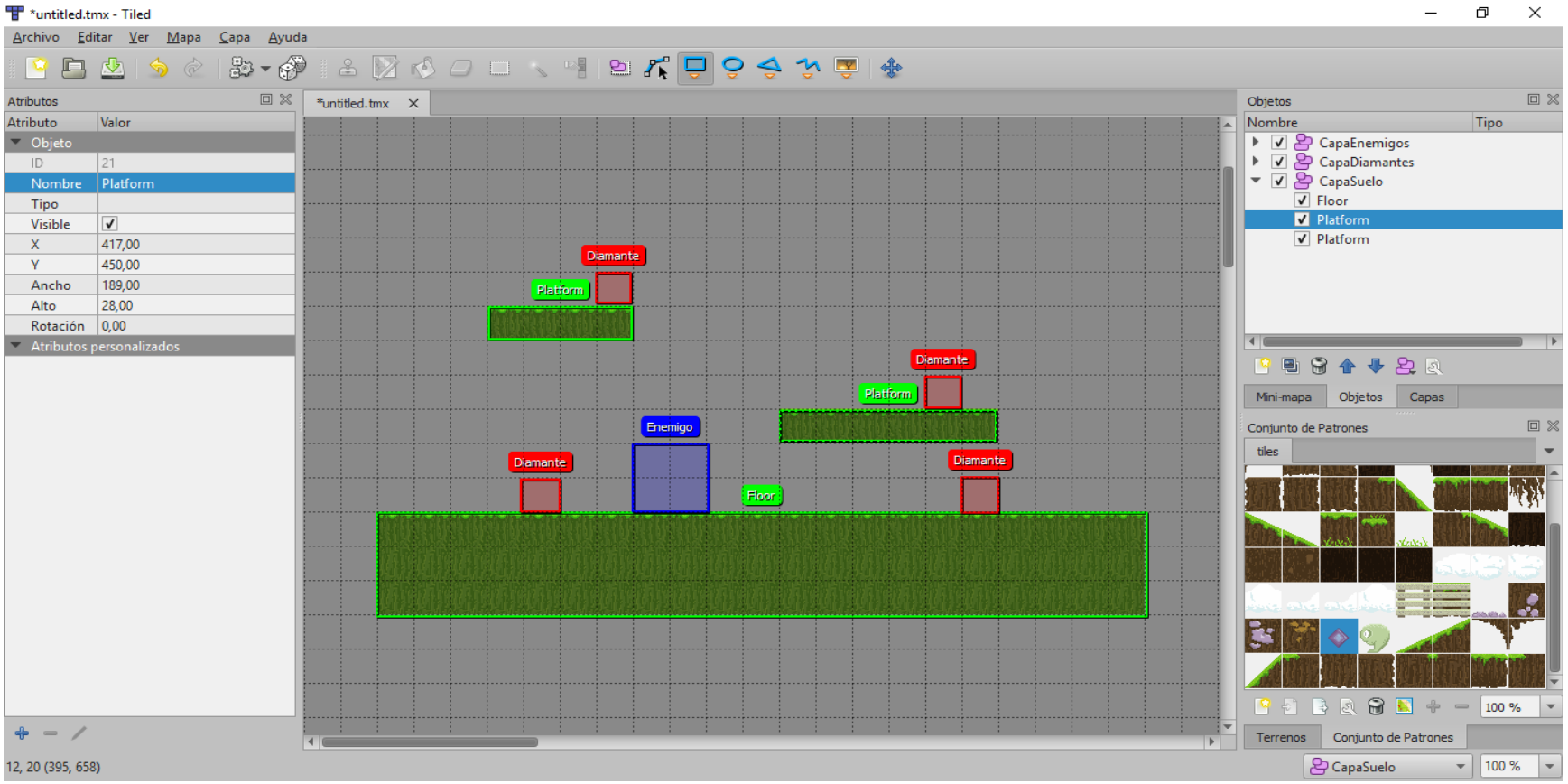
USING TILED

Creating A Map

- The layer options allow to create an object layer. We can draw the shapes of the objects the player can interact with like a platforms, enemies or items.
- This option works perfectly with Box2D.
- Some of this interactive elements, like enemies, are not in the tileset. We can indicate where they are but we need to implement them.

USING TILED

Creating A Map



IMPLEMENTING THE MAP IN OUR GAME

Using LibGDX

- We need the *.tmx file we got by saving the map with Tiled.
- Libgdx doesn't directly support tmx so we have to take some steps first.
- First we will need to tell the assets manager to use the TmxMapLoader class when loading a TiledMap object.

```
public void create() {

 assetManager.setLoader(TiledMap.class,
                        new TmxMapLoader(new InternalFileHandleResolver()));

 setScreen(new LoadingScreen(this));

}
```

IMPLEMENTING THE MAP IN OUR GAME

Using LibGDX

- Once in the Loading Screen we will load the file itself.
- As we have told the Asset Manager how to deal with this type of files it will know what to do.

```
public void show() {

     ...
     myGame.getAssetManager().load("map.tmx", TiledMap.class);
     ...
}
```

IMPLEMENTING THE MAP IN OUR GAME

Using LibGDX

- Now, after loading it, we can work with it in our GameScreen class.
- We will need a TiledMap object to get the map.
- We will also need an OrthogonalTiledMapRenderer object to render it.

```
private TiledMap tiledMap;

public void show() {
     ...
     tiledMap = peteGame.getAssetManager().get("map.tmx");

     orthogonalTiledMapRenderer =
                  new OrthogonalTiledMapRenderer(tiledMap, batch);

     orthogonalTiledMapRenderer.setView(camera);
     ...
}
```

IMPLEMENTING THE MAP IN OUR GAME

Using LibGDX

```
Using LibGDX
private void draw() {

     batch.setProjectionMatrix(camera.projection);
     batch.setTransformMatrix(camera.view);
     orthogonalTiledMapRenderer.render();
}
```